



## Unidad 03: Tipo de dato primitivo - operadores

2

### Contenidos analíticos

#### Parte I: Del Universo real al universo computacional

##### Unidad 3: Tipo de dato primitivo y operadores

Definición de tipo de dato. Tipos de datos primitivos (definición de los conjuntos de valores y operaciones aplicadas a los mismos. Definición y alcance de tipos de datos simples, ordinales y no ordinales, introducción al concepto de cadenas de caracteres. Operadores lógicos, aritméticos de relación de asignación, utilización y combinaciones de los mismos. Patrones algorítmicos con datos simples: secuencias, seguidillas, selecciones.

### Tipos de datos

Los tipos de datos Identifican o determinan un dominio de valores y un conjunto de operaciones aplicables sobre esos valores.

1. Primitivos.
2. Derivados.
3. Abstractos.

Los algoritmos operan sobre datos de distinta naturaleza, por lo tanto los programas que implementan dichos algoritmos necesitan una forma de representarlos.

Tipo de dato es una clase de objeto ligado a un conjunto de operaciones para crearlos y manipularlos, un tipo de dato se caracteriza por

1. Un rango de valores posibles.
2. Un conjunto de operaciones realizadas sobre ese tipo.
3. Su representación interna.

Al definir un tipo de dato se esta indicando los valores que pueden tomar sus elementos y las operaciones que pueden hacerse sobre ellos.

Al definir un identificador de un determinado tipo el nombre del identificador indica la localización en memoria, el tipo los valores y operaciones permitidas, y como cada tipo se representa de forma distinta en la computadora los lenguajes de alto nivel hacen abstracción de la representación interna e ignoran los detalles pero interpretan la representación según el tipo.

Tipos de datos pueden ser.

1. **Estáticos:** Ocupan una posición de memoria en el momento de la definición, no la liberan durante el proceso solamente la liberan al finalizar la aplicación.



- a. **Simples:** Son indivisibles en datos mas elementales, ocupan una única posición para un único dato de un único tipo por vez.
  - i. **Ordinales:** Un tipo de dato es ordinal o esta ordenado discretamente si cada elemento que es parte del tipo tiene un único elemento anterior (salvo el primero) y un único elemento siguiente (salvo el ultimo).
    - 1. **Enteros:** Es el tipo de dato numérico mas simple.
    - 2. **Lógico** o booleano: puede tomar valores entre dos posibles: verdadero o falso.
    - 3. **Carácter:** Proporcionan objetos de la clase de datos que contienen un solo elemento como valor. Este conjunto de elementos esta establecido y normatizado por el estándar ASCII.
  - ii. **No ordinales:** No están ordenados discretamente, la implementación es por aproximación
    - 1. **Reales:** Es una clase de dato numérico que permite representar números decimales.
- b. **Cadenas:** Contienen N caracteres tratados como una única variable.
- c. **Estructuras:** Tienen un único nombre para mas de un dato que puede ser del mismo tipo o de tipo distinto. Permiten acceso a cada dato particular y son divisibles en datos mas elementales.

Una estructura es, en definitiva, un conjunto de variables no necesariamente del mismo tipo relacionadas entre si de diversas formas.

Si los datos que la componen son todas del mismo tipo son homogéneas, heterogéneas en caso contrario.

Una estructura es estática si la cantidad de elementos que contiene es fija, es decir no cambia durante la ejecución del programa

- i. **Registro:** Es un conjunto de valores que tiene las siguientes características:
  - Los valores pueden ser de tipo distinto. Es una estructura heterogénea.
  - Los valores almacenados se llaman campos, cada uno de ellos tiene un identificador y pueden ser accedidos individualmente.
  - El operador de acceso a cada miembro de un registro es el operador punto.
  - El almacenamiento es fijo.

- ii. **Arreglo:** Colección ordenada e indexada de elementos con las siguientes características:

Todos los elementos son del mismo tipo, un arreglo es una estructura homogénea.

Los elementos pueden recuperarse en cualquier orden, simplemente indicando la posición que ocupa dentro de la estructura, esto indica que el arreglo es una estructura indexada.

El operador de acceso es el operador []

La memoria ocupada a lo largo de la ejecución del programa es fija, por esto es una estructura estática.

El nombre del arreglo se socia a un área de memoria fija y consecutiva del tamaño especificado en la declaración.

El índice debe ser de tipo ordinal. El valor del índice puede verse como el desplazamiento respecto de la posición inicial del arreglo.

Los arreglos pueden ser de varias dimensiones. Esta dimensión indica la cantidad de índices necesarias para acceder a un elemento del arreglo.

El arreglo lineal, con un índice, o una dimensión se llama vector.



El arreglo con 2 o mas índices o dimensiones es una matriz. Un grupo de elementos homogéneo con un orden interno en el que se necesitan 2 o mas índices para referenciar a un elemento de la estructura.

- iii. **Archivos:** Estructura de datos con almacenamiento físico en memoria secundaria o disco.

Las acciones generales vinculadas con archivos son

Asignar, abrir, crear, cerrar, leer, grabar, Cantidad de elementos, Posición del puntero, Acceder a una posición determinada, marca de final del archivo, definiciones y declaraciones de variables.

Según su organización pueden ser secuenciales, indexados.

1. **Archivos de texto:** Secuencia de líneas compuestas por cero uno o mas caracteres que finalizan con un carácter especial que indica el final de la línea. Los datos internos son representados en caracteres, son mas portables y en general mas extensos.
2. **Archivos de tipo o binarios:** secuencia de bytes en su representación interna sin interpretar. Son reconocidos como iguales si son leídos de la forma en que fueron escritos. Son menos portables y menos extensos.
2. **Dinámicos:** Ocupan direcciones de memoria en tiempo de ejecución y se instancian a través de punteros. Estas instancias pueden también liberarse en tiempo de ejecución. El tema de puntadores y estructuras enlazadas (estructuras relacionadas con este tipo de dato se analizan en detalle en capítulos siguientes)
  - a. **Listas simplemente enlazadas:** cada elemento sólo dispone de un puntero, que apuntará al siguiente elemento de la lista o valdrá NULL si es el último elemento.
  - b. **Pilas:** son un tipo especial de lista, conocidas como listas LIFO (Last In, First Out: el último en entrar es el primero en salir). Los elementos se "amontonan" o apilan, de modo que sólo el elemento que está encima de la pila puede ser leído, y sólo pueden añadirse elementos encima de la pila.
  - c. **Colas:** otro tipo de listas, conocidas como listas FIFO (First In, First Out: El primero en entrar es el primero en salir). Los elementos se almacenan en fila, pero sólo pueden añadirse por un extremo y leerse por el otro.
  - d. **Listas circulares:** o listas cerradas, son parecidas a las listas abiertas, pero el último elemento apunta al primero. De hecho, en las listas circulares no puede hablarse de "primero" ni de "último". Cualquier nodo puede ser el nodo de entrada y salida.
  - e. **Listas doblemente enlazadas:** cada elemento dispone de dos punteros, uno a punta al siguiente elemento y el otro al elemento anterior. Al contrario que las listas abiertas anteriores, estas listas pueden recorrerse en los dos sentidos.
  - f. **Árboles:** cada elemento dispone de dos o más punteros, pero las referencias nunca son a elementos anteriores, de modo que la estructura se ramifica y crece igual que un árbol.
  - g. **Árboles binarios:** son árboles donde cada nodo sólo puede apuntar a dos nodos.
  - h. **Árboles binarios de búsqueda (ABB):** son árboles binarios ordenados. Desde cada nodo todos los nodos de una rama serán mayores, según la norma que se haya seguido para ordenar el árbol, y los de la otra rama serán menores.
  - i. **Árboles AVL:** son también árboles de búsqueda, pero su estructura está más optimizada para reducir los tiempos de búsqueda.
  - j. **Árboles B:** son estructuras más complejas, aunque también se trata de árboles de búsqueda, están mucho más optimizados que los anteriores.



- k. **Tablas HASH:** son estructuras auxiliares para ordenar listas.
- l. **Grafos:** es el siguiente nivel de complejidad, podemos considerar estas estructuras como árboles no jerarquizados.
- m. **Diccionarios.**

### Criterios de selección

1. Priorizar de ser posible acceso directo y velocidad de procesamiento.
  - a. Vector en primer lugar si se cumple tamaño fijo, razonable y conocido a priori, y sin necesidad de persistencia → acceso directo, búsqueda binaria, búsqueda secuencial
2. Si no se conoce el tamaño y no se requiere persistencia
  - a. Estructuras enlazadas
    - i. Pila si se debe invertir el orden o si es irrelevante
    - ii. Colas si se debe mantener
    - iii. Listas si se debe generar
3. Archivo si se requiere persistencia: directamente o estructuras auxiliares después cargar al archivo

### Toma de decisiones

1. Los datos se ingresan desde el teclado
  - a. Se deben mostrar en la misma secuencia de entrada: No es necesario guardarlos en una estructura auxiliar. Así como los recibimos se deben mostrar, no se requiere conservarlos en memoria para ningún procesamiento posterior.
  - b. Se requiere procesamiento posterior. Supongamos que debemos mostrarlos ordenados por un criterio diferente al ingreso: en este caso se debe generar una estructura auxiliar, esta puede ser un vector o lista si el ordenamiento es por un campo, matriz, vector de listas o lista de listas si el ordenamiento es por dos criterios
2. Los datos se ingresan desde un archivo físico
  - a. Similar al ingreso por teclado → solo cambia el origen del dato
  - b. Similar al ingreso por teclado → solo cambia el origen del dato
3. Secuencia de decisiones
  - a. Origen del dato
    - i. Teclado
    - ii. Archivo
  - b. Elección de estructura auxiliar
    - i. Solo se muestra con el criterio de ingreso; no se requiere estructura auxiliar
    - ii. Se requiere reordenar o conservar para buscar: Requiere auxiliar
      1. Orden por un campo el tamaño es conocido a priori
        - a. Vector
          - i. Carga directa
          - ii. Carga secuencial
            1. Ordenar posteriormente
            2. Cargar ordenada
            3. Dejar sin orden, búsqueda secuencial
      2. Orden por un campo tamaño no conocido
        - a. Lista ordenada
          - i. Insertar ordenado
          - ii. CargarSinRepetir



- iii. InsertarOrdenado
- 3. Orden por dos campos
  - a. Ambos conocidos y acotados
    - i. Vector de vector → matriz
  - b. Uno acotado y definido, el otro no
    - i. Vector de punteros
  - c. Ambos No acotados
    - i. Lista de listas
- c. Que guardar en la estructura auxiliar
  - i. Todos los datos si se requieren
  - ii. Solo los que se requieren como salida, no hacer substancia de lo superfluo
  - iii. La clave de ordenamiento y la referencia al dato

### Tipos de datos simples – Tamaño y Rango

Nombre del Tipo	Otros Nombres	Implementaciones					
		Borland Turbo C++ 3.0 16 bits		Borland C++ 4.02 32 bits		Microsoft Visual Studio .NET 32 bits	
		Bytes	Rango	Bytes	Rango	Bytes	Rango
<b>Char</b>	signed char	1	-128 .. 127 [ $2^7$ .. ( $2^7-1$ )] (ASCII Standard)				
<b>unsigned char</b>	-	1	0 .. 255 [ $2^8$ .. ( $2^8-1$ )] (ASCII Extendido)				
<b>Short</b>	short int, signed short, signed short int	2	-32,768 .. 32,767 [ $2^{15}$ .. ( $2^{15}-1$ )]				
<b>unsigned short</b>	unsigned short int	2	0 .. 65,535 [ $2^{16}$ .. ( $2^{16}-1$ )]				
<b>Int</b>	Signed, signed int	2	-32,768 .. 32,767 [ $2^{15}$ .. ( $2^{15}-1$ )]	4	-2,147,483,648 .. 2,147,483,647 [ $2^{31}$ .. ( $2^{31}-1$ )]		
<b>unsigned int</b>	unsigned	2	0 .. 65,535 [ $2^{16}$ .. ( $2^{16}-1$ )]	4	0 .. 4,294,967,295 [ $2^{32}$ .. ( $2^{32}-1$ )]		
<b>Long</b>	long int, signed long, signed long int	4	-2,147,483,648 .. 2,147,483,647				
<b>unsigned long</b>	unsigned long int	4	0 .. 4,294,967,295				
<b>Enum</b>	-		<i>igual a int</i>				
<b>Float</b>	-	4	$3.4 \times 10^{-38}$ .. $3.4 \times 10^{38}$ (7 dígitos de precisión)				
<b>Double</b>	-	8	$1.7 \times 10^{-308}$ .. $1.7 \times 10^{308}$ (15 dígitos)				
<b>long double</b>	-	10	$3.4 \times 10^{-4932}$ .. $1.1 \times 10^{4932}$ (19 dígitos)				<i>igual a double</i>



## Precedencia y Asociatividad de los 45 Operadores

ID Asociatividad Izquierda-Derecha, DI Asociatividad Derecha-Izquierda

1	ID	operadores de acceso	( )		invocación a función
			[ ]		subíndice de arreglo
			.		acceso a struct y a union
			->		acceso a struct y a union
2	DI	operadores unarios (operan sobre un solo operando)	+	-	signos positivo y negativo
			~		complemento por bit
			!		NOT lógico
			&		dirección de
			*		"indirección"
			++		pre-incremento
			--		pre-decremento
			sizeof		tamaño de
			(tipo)		conversión explícita
3	ID	operadores multiplicativos	*		multiplicación
			/		cociente
			%		módulo o resto
4	ID	operadores aditivos	+	-	suma y resta
5	ID	operadores de desplazamiento izquierda	<<		desplazamiento de bits a izquierda
		derecha	>>		desplazamiento de bits a derecha
6	ID	operadores relacionales	<	>	
			<=	>=	
7	ID	operadores de igualdad	==	!=	igual a y distinto de
8	ID	operadores binarios por bit	&		AND
9	ID		^		OR exclusivo
10	ID				OR
11	ID	operadores binarios lógicos	&&		AND
12	ID				OR
13	DI	operador condicional (operandos)	?:		(único que opera sobre 3 operandos)
14	DI	operadores de asignación	=		
			*=	/=	%=
			<<=	>>=	&=
					+=
					-=
					^=
					=
15	ID	operador concatenación expresiones	,		"coma"
		Los operadores &&,    y "coma" son los únicos que garantizan que los operandos sean evaluados en un orden determinado (de izquierda a derecha).			
		El operador condicional (?:) evalúa solo un operando, entre el 2do. y el 3ro., según corresponda.			



## Algunas consideraciones

El resultado de la división de dos enteros es un entero, la división de enteros es, entonces una división entera, por tanto siendo:

Int a=10, b=2, c=3;

float d = 3.0, e=10.0;

Determine los resultados de

a/b

a/c

a/3.0

a/d

e/d

Justifique la respuesta → Ayuda: determine que es jerarquía de datos, cual es la secuencia (char, int, long, float, double), que pasa en una expresión con datos de diferentes jerarquías, busque la definición casteo explícito, implícito

Es posible asignar a=b=c=15; que valores asigna a que identificadores, porque, es la asignación = un operador o una sentencia?

Siendo a=10; que efecto produce a++; y si hubiera sido ++a? que hubiera pasado con –

Siendo int a = 10, b;

¿Que valores toman a y b luego de b=a++;? ¿Por qué?

Siendo int a = 10, b;

¿Que valores toman a y b luego de b = ++a?; ¿Por qué?

Que valor toman a y b luego de a+=b; y a\*=b, Por qué?

Sea int a;

Es posible la asignación a = 'A'+ 'B'; esta sentencia le asigna valor al identificador a?, ¿que valor?

¿Por que?



## Ejercicios

- 1.Cuál de las siguientes sentencias son correctas para la ecuación algebraica  $y=ax^3+7$ .
  - a.  $y = a * x * x * x + 7$
  - b.  $y = a * x * x * (x + 7)$
  - c.  $y = (a * x) * x * (x + 7)$
  - d.  $y = (a * x) * x * x + 7$
  - e.  $y = a * (x * x * x) + 7$
  - f.  $y = a * (x * x * x + 7)$
2. Escriba un programa que pida al usuario dos números e informe la suma, la resta, el producto y el cociente de los mismos
3. Imprima un programa que imprima los números del 1 al 4 en una misma línea, hágalo de las formas siguientes:
  - a. Utilizando un solo operador de inserción de flujo
  - b. Una única sentencia con 4 operadores de inserción de flujo
  - c. Utilizando cuatro sentencias
4. Escriba un programa que reciba tres números por el teclado e imprima la suma, el promedio, el producto, el mayor y el menor de esos números. Escriba un adecuado dialogo en pantalla.
5. Escriba un programa que reciba un numero que represente el radio de un círculo e imprima el diámetro, circunferencia y área.
6. Que imprime el siguiente código
  - a. `std::cout << "\n**\n***\n****" <<std::endl;`
  - b. `std::cout << 'A';`
  - c. `std::cout << static_cast<int> 'A';` (que es static\_cast? Investigue.)
7. Utilizando solo lo que hemos desarrollado en esta introduccion escriba un programa que calcule los cuadrados y los cubos de los números de 0 a 10 y los muestre por pantalla.
8. Escriba un programa que reciba un numero entero de 5 digitos, que separe el numero en sus digitoe y los muestre por pantalla, uno por línea comenzando por elmas significacivo en la primera línea.
9. Dados dos valores enteros y positivos determinar y mostrar por el dispositivo estándar de salida: la suma, la resta, el producto y la división de los mismos. Analice precondiciones adecuadas y utilice leyendas adecuadas. Resuelva teniendo en cuenta las precondiciones y da una solución alternativa sin considerarlas.
10. Responda: que pasa si el conjunto de datos es float, si es entero y sin la restricción de ser positivo.
11. Dada una terna de números naturales <dia, mes, año> que representan al día, al mes y al año de una fecha informarla como un solo número natural de 8 dígitos (AAAAMMDD). Establezca precondiciones.
12. Dada un número natural de 8 dígitos, con formato (AAAAMMDD) descompóngalo en sus elementos lógicos y muéstrellos por el dispositivo estándar de salida.
13. Dado un valor entero determinar y mostrar: la quinta parte del mismo, el resto de la división por 5 y la tercera parte del valor del primer punto. Resuelva sin precondiciones, informando por la salida estándar los resultados.
14. Dado una terna de valores determine e imprima: El mayor y el menor del conjunto. Muestre por salida estándar con las aclaraciones que considere.
15. Que cambios estratégicos produciría si el conjunto de valores ingresados fueran 20? Y si el lote fuera de una cantidad no conocida a priori?





16. Dadas dos ternas de valores que representan las fechas validas de nacimiento de dos personas, indique cual de las dos corresponde al mayor. Utilice las leyendas que crea corresponden.
17. Dado un par de valores que representa una el mes y año de una fecha valida del siglo XXI determinar e imprimir la cantidad de días de ese mes.
18. Dado una terna de valores enteros y positivos determinar si los mismos forman un triángulo.
19. Dado una terna de valores que representan los lados de un triangulo determinar si el tipo de triangulo que forman.
20. Dados dos números naturales M y N, determinar e imprimir cuantos múltiplos de M hay el el conjunto 1 a N.
21. Dados dos números enteros, M y N informar su producto por sumas sucesivas.
22. Dados un conjunto de valores enteros informar el promedio de los mayores que 45 y la suma de los menores que -10.
23. Dado un valor M determinar e imprimir los M primeros múltiplos de 3 que no lo sean de 5, dentro del conjunto de los números naturales.
24. Dados un conjunto de valores enteros y positivos determinar e informar el mayor
25. Dado un conjunto de N valores informar el mayor, el menor y en qué posición del conjunto fueron ingresados.
26. Dado un conjunto de valores reales, que finaliza con un valor nulo, determinar e imprimir (si hubieron valores): el máximo de los negativos y el minimo de los positivos



Con la unión de la documentación referida a las unidades 01, 02 y 03 Usted debería estar en condiciones de:

**Logros Pedagógicos:** Comprender los problemas de información con la utilización de datos de tipo simple y utilizar el concepto de refinamientos sucesivos para la eficaz resolución de un problema computacional Mediante el desarrollo conceptual y/o gráfico del algoritmo más adecuado para la resolución.

**Cuestionario de Autoevaluación de la primera parte del Universo real al universo computacional**

1. ¿En qué área y bloque ubica a la materia Algoritmos y estructura de datos del departamento DISI de la UTN.BA?
2. Señale logros de aprendizaje o competencias específicas que espera de la cursada de la materia.
3. Puede determinar qué porcentaje de la formación total de la carrera del ingeniero en sistemas corresponden a horas del área de programación? Y la materia AyED dentro del área y la carrera en su totalidad?
4. Qué coincidencias, diferencias y relaciones encuentra entre las disciplinas computación, informática y programación?
5. Describa cuál es el eje troncal de la carrera Ingeniero en sistemas de información de DISI UTN.BA
6. En qué fecha, año, mes y día, estima se recibirá de Ingeniero en Sistemas de UTN.BA?
7. ¿Cuál entiende es el propósito de Algoritmos y estructura de datos?
8. Con qué otras materias las vincula y cuál cree que son las relaciones entre ellas?
9. ¿Qué entiende por paradigma? Paradigma de programación
10. Describa los paradigmas de programación que conoce y alguna particularidad de los mismos
11. ¿Qué entiende por dato? Se diferencia de información?
12. Describa brevemente los conceptos de: programación, programa, proceso, tarea. Así como la relación entre los mismos si es que las tienen.
13. ¿Qué entiende por precondition? Ejemplifique si se le ocurre definir alguna en caso de tener que resolver la situación problemática de *Dados dos números determinar y mostrar la suma, resta, multiplicación y cociente*.
14. Defina algoritmo, sus características y propiedades
15. ¿Qué entiende por identificador. Ejemplifique
16. ¿Qué entiende por estrategia?
17. ¿Cómo definiría refinamientos sucesivos
18. Establezca diferencia entre los siguientes pares de conceptos (declaración, definición) , (expresión, sentencia); (constantes, variables)
19. Describa las etapas necesarias para la resolución de problemas de información.
20. ¿Qué diferencia observa entre los conceptos de notación algorítmica y representación gráfica para la resolución de los algoritmos
21. Describa las representaciones gráficas que conoce,
22. ¿Qué entiende por proceso computacional